

# Google Glass en logistique

Rapport intermédiaire de PFE

---

**Tuteur école : Laurent Grisoni**  
**Tuteur entreprise: Laurent Borel**

**Vincent Meunier**  
**Jérémy Gondry**  
**IMA5 promo 2015**



## Sommaire

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Présentation Entrepôt/Problématique .....	3
1.2	Cahier des Charges.....	4
<b>2</b>	<b>Etude du Besoin.....</b>	<b>5</b>
2.1	Réalité métier.....	5
2.2	Possibilités et limites des SmartGlass .....	5
<b>3</b>	<b>Travail réalisé .....</b>	<b>7</b>
3.1	Prise en main des SmartGlass .....	7
3.1.1	Prise en main des GoogleGlass .....	7
3.1.2	Prise en main des ORA de Optinvent.....	7
3.2	Algorithme de recherche .....	7
3.3	Interaction et Interface .....	10
3.3.1	Reconnaissance vocale .....	10
3.3.2	Scénario d'utilisation.....	11
<b>4</b>	<b>Ce qui reste à faire.....</b>	<b>13</b>
4.1	Développement des SmartGlass .....	13
4.2	Projet d'affichage personnalisé TV .....	13
<b>5</b>	<b>Conclusion.....</b>	<b>14</b>
5.1	Essai de l'application.....	14
5.2	Bilan .....	14

# 1 Introduction

## 1.1 Présentation Entrepôt/Problématique

Notre projet vise à l'optimisation du stockage et de l'adressage des palettes dans un entrepôt. En effet, lors du déchargement d'un camion, chercher un emplacement libre pour les palettes dans un entrepôt occupant une grande superficie fait perdre du temps et donc de la productivité dans le stockage. Sachant qu'un cariste perd un temps non négligeable à la recherche d'emplacements disponibles, optimiser cette tâche de recherche permettrait de gagner du temps sur cette opération.

Notre projet consiste donc à fluidifier le déchargement en indiquant au cariste l'emplacement libre le plus proche.

Développer un prototype permettrait de rendre compte des moyens existants qui pourrait être mis en place dans une solution métier.



*Google Glass*



*ORA d'Optivent*

Pour ne pas gêner le travail des employés, un dispositif main-libre comme une paire de Google Glass semble adapté. Ce projet soulève plusieurs problématiques : développer une interface homme-machine apportant une information pertinente pour l'utilisateur sans pour autant être perturbante.

L'idée serait de réaliser dans les grandes lignes un système similaire à ce scénario d'utilisation : <http://n.mynews.ly/!B.Bs4fw>

## 1.2 Cahier des Charges

En partant de l'analyse du sujet et des discussions avec notre tuteur, nous avons défini un cahier des charges plus précis des tâches à accomplir:

- **Développer une application embarquée sur une paire de lunette Smart Glass**
  - Comparer les différentes offres en matière de Smart Glass notamment Google Glass & ORA de Optinvent  
*Les Google Glass ne sont pas les seules smart glasses sur le marché. Il nous faudra choisir les appareils les plus adaptés.*
  - Concevoir une interface homme-machine non perturbante pour l'utilisateur.
  - Proposer un système de reconnaissance vocale en français pour l'interaction main libre avec l'utilisateur.
- **Etudier la nécessité d'utiliser une géolocalisation en entrepôt et l'implémenter si besoin**  
*L'information pertinente pourrait être une direction à suivre dans l'entrepôt ou simplement un numéro d'emplacement (l'adressage de l'entrepôt est intuitif).*
  - Via RSSI (Wifi, Bluetooth Low Energy ...)
  - Via traitement d'images (reconnaissance de la signalétique en entrepôt)
- **Etudier les possibilités d'optimisation du système d'adressage des palettes**  
*En ce qui concerne l'adressage des palettes en entrepôt, la solution existante repose sur un flashage des codes barres associés à l'emplacement venant d'être occupé. Une étude de faisabilité/utilité sera aussi faite sur une solution de remplacement grâce aux possibilités qu'offrent les smart glasses.*
  - Implémenter la reconnaissance de code barre
  - La tester en réalité métier (utilité ajoutée, habitude des magasiniers, vitesse de traitement...)

## 2 Etude du Besoin

### 2.1 Réalité métier

Tout au long de notre projet il nous aura été demandé de réfléchir sur nos réalisations et de prendre du recul par rapport à notre travail. En effet, le concept de réalité métier a souvent été abordé pour savoir si un magasinier pourrait réellement se servir de notre application, et si cela lui apporte une vraie plus valu dans son travail. Dans cette optique de réflexion plusieurs sujets ont été abordés:

- Une interface graphique peu intrusive
- Une information pertinente et simple à récupérer
- Un système main libre qui ne gêne pas l'utilisateur

Ce sont ces trois axes qui nous ont guidés lors de la création de l'interface des Google Glass.

### 2.2 Possibilités et limites des SmartGlasses

	Google Glass	ORA
<b>Connectique</b>	Compatible BLE (à partir de la XE 16 KitKat Update), Wifi	Compatible BLE, Wifi
<b>Autonomie en utilisation normale</b>	de 1h à 2h	de 4h à 8h
<b>Son</b>	Conductivité Cranienne	Prise jack
<b>Interface</b>	Ecran occupant une faible superficie dans le champ de vision	Ecran 2 positions (moyennes et basses), plus étendu permettant une réalité augmentée
<b>Système d'exploitation</b>	Android 4.4	Android 4.2.2
<b>Calculs</b>	CPU & GPU	CPU & GPU
<b>Prix</b>	1500\$	949\$

Les Google Glass sont bien plus chères que les ORA. De plus les ORA permettent d'avoir une réalité augmentée et sont dotées d'une bien plus grande autonomie. Cependant après avoir testé le prototype des lunettes ORA et assisté à une conférence à l'Imaginarium de Tourcoing sur les Google Glass, il en est ressorti que les Google Glass seraient finalement plus adaptée pour notre projet, l'affichage des ORA est trop présent, et il faut la mettre en position basse pour ne pas être perturbé et le produit n'est pas encore fini, de plus le poids est mal réparti, ce déséquilibre dérangerait forcément un magasinier sur son chariot. A l'inverse, les Google Glass sont plus légères, plus équilibrées. Et permettent d'obtenir des informations dans un coin du champ de vision non gênant.

Des applications sur les Google Glass pour la médecine ont été développées (leur efficacité d'utilisation main libre peut se faire une place en milieu stérile), également pour les pompiers (une application qui affiche pour un modèle de voiture accidentée donné, la procédure de désossage pour en extirper un corps, les moyens actuels reposant sur des gros manuels).

## 3 Travail réalisé

### 3.1 Prise en main des SmartGlass

#### 3.1.1 Prise en main des GoogleGlass

L'utilisation de la Mirror API permet à des développeurs de coder des applications dans le langage que l'on souhaite, qui seraient exécutées sur les serveurs de Google et seraient téléchargeables par d'autres utilisateurs. Les serveurs s'occuperaient des échanges d'informations, beaucoup d'outils sur les lunettes fonctionnent sur ce principe.

Le GDK, une surcouche du SDK, permet d'avoir une emprise sur le matériel des lunettes (communication Bluetooth, accéléromètre, etc... ) via des méthodes et objets spécifiques à celui-ci. Le développement se basant sur le GDK permet une instantanéité de l'application.

C'est la raison pour laquelle nous avons choisi de travailler avec le GDK.

Nous avons rencontré soucis de drivers, le problème venant de l'association entre les ID des lunettes et ceux qui sont renseignés dans le fichier "AndroidWinusb" après modification du fichier, il a fallu autoriser notre système d'exploitation sur lequel nous développons pour autoriser l'utilisation de pilote dont la signature numérique est erronée.

Après ce problème réglé nous avons pris en main les fonctions du GDK en programmant des interactions simple (vocal, touchpad) pour mieux s'imprégner du développement sur cette appareil.

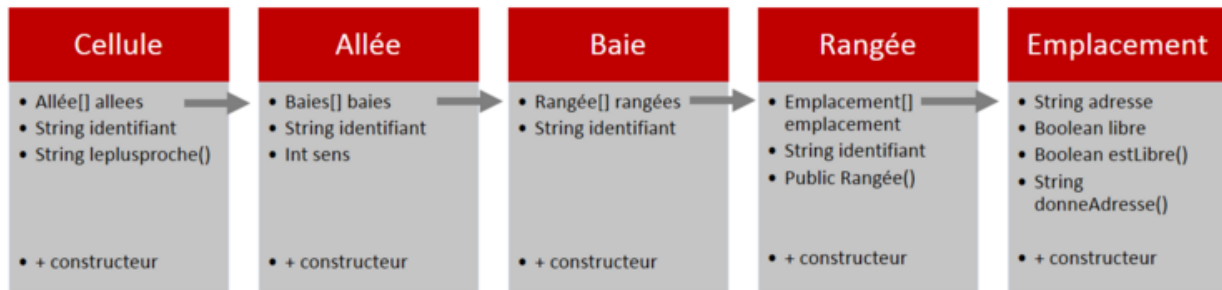
#### 3.1.2 Prise en main des ORA de Optinvent

Ces lunettes possèdent un système Android 4.2.2 Avec une interface graphique classique. Le touchpad permet de déplacer le curseur sur l'écran et de sélectionner/lancer une application. Nous avons rencontré un problème de drivers également, il nous a fallu demander au support technique d'Optinvent, les drivers pour notre système d'exploitation.

Une fois le problème de driver résolu nous n'avons eu aucun souci à charger un programme et à le lancer sur les lunettes.

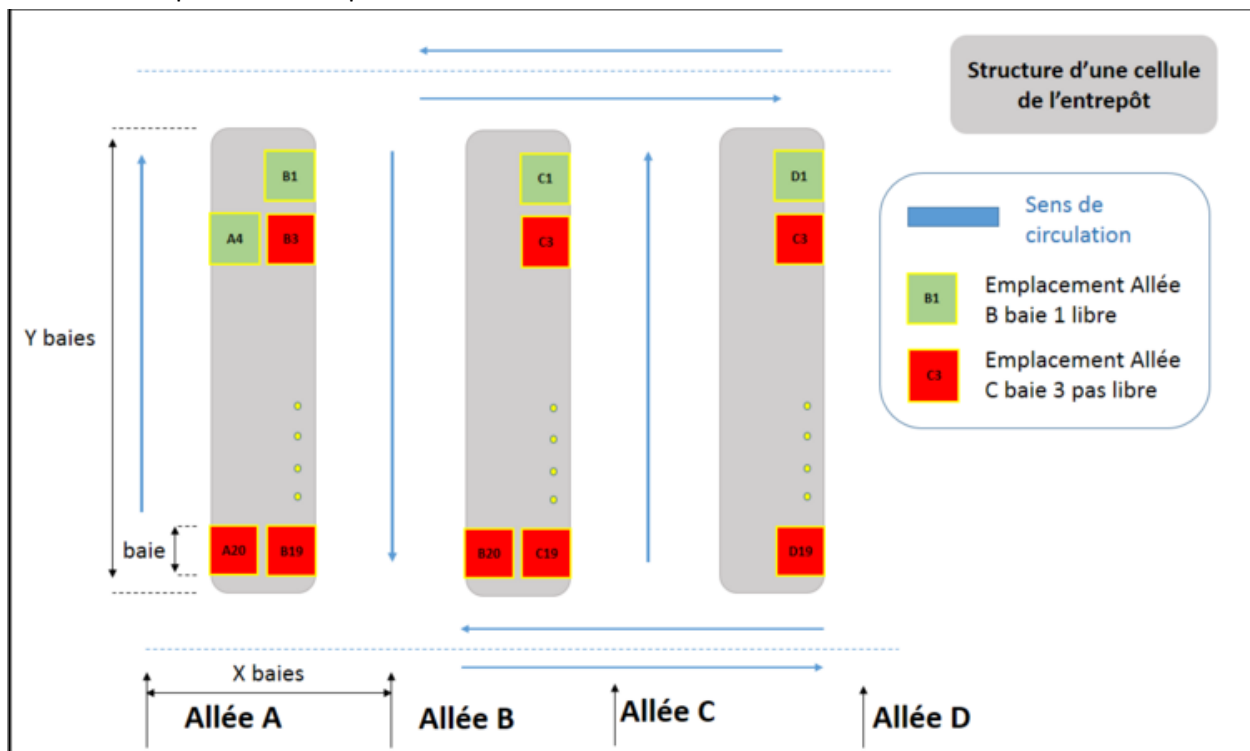
### 3.2 Algorithme de recherche

Choix technique : Dans un premier temps la recherche se base sur l'analyse d'une structure de donnée en dur dans l'application. La communication avec la base de données n'étant pour le moment pas une priorité pour l'entreprise, et pour tester notre algorithme de recherche, nous avons opté pour cette structure de donnée récursive.



Nous avons écrit une première méthode de détermination. On choisit l'emplacement, parmi tous ceux qui sont libres, pour lequel la distance de parcours est minimale entre le départ et celui-ci. Ici l'unité de distance est la baie. La simplicité de l'algorithme réside dans le fait que le magasinier ne transporte qu'une seule caisse à la fois et repars toujours au même point de départ. C'est une sélection intelligente des adresses et non un parcours intelligent des adresses (l'emplacement libre trouvé le plus tôt n'est pas forcément le plus proche).

Ci dessous un plan de l'entrepôt.



On fixe une distance entre les allées (X baies) et une longueur d'allée (Y baies). L'entrepôt étant récent et assez homogène dans sa topologie.

Si on se place à l'entrée de l'Allée A :



Le déplacement jusqu'à A4 (sens de circulation direct) est de :

$$Y \text{ baies} - (\text{indice de la baie}/2 + \text{indice de la baie} \% 2)$$

Le déplacement jusqu'à B1 (sens de circulation indirect) est de :

$$Y \text{ baies} + \text{différence entre l'allée A et B donc } X \text{ baies} + \text{indice de la baie}/2 + \text{indice de la baie} \% 2$$

Le déplacement jusqu'à C1 (sens de circulation direct) est de :

$$Y \text{ baies} - (\text{indice de la baie}/2 + \text{indice de la baie} \% 2) + \text{diff entre l'allée A et C donc } 2 * X \text{ baies.}$$

Si on se place à l'entrée de l'Allée B :

Le déplacement jusqu'à B1 (sens de circulation indirect avec contournement) est de :

$$2 * X \text{ baies} + Y \text{ baies} + 2 * X \text{ baies} + (\text{indice de la baie}/2 + \text{indice de la baie} \% 2)$$

On peut généraliser l'ensemble de ces situations de déplacement avec l'algorithme suivant

Leplusproche( Allee depart )

distance\_la\_plus\_petite = valeurMAX

le\_plus\_proche=""

Pour chaque Allee de Cellule

sens=Allee.sens

Pour chaque Baie de Allee

Pour chaque Rang de Allee

Pour chaque Emplacement de Rang

Si Emplacement est libre

indice\_baie = selection\_indice\_baie(Emplacement.adresse)

allee\_cible = selection\_indice\_allee(Emplacement.adresse)

différence\_allee = valeur\_absolue(depart - cible)

Si différence\_allee==0 et sens.estindirect

//sens indirect +contournement

distance\_actuelle=(indice\_baie/2)+indice\_baie%2+(2\*(différence\_allee))+4\*X+Y

Sinon

Si sens est direct

//sens direct

distance\_actuelle=(Y-((indice\_baie/2)+indice\_baie%2))+2\*différence\_allee

Sinon

//sens indirect mais pas de contournement

distance\_actuelle=((indice\_baie/2)+indice\_baie%2)+2\*différence\_allee+Y

Finsi

```

    Finsi
    Si distance_actuelle < distance_la_plus_petite
        le_plus_proche=Emplacement.adresse
        distance_la_plus_petite=distance_actuelle
    Finsi
Finsi
    Finpour
    Finpour
    Finpour
    Finpour
    Finpour
    retourne le_plus_proche
Finleplusproche

```

### 3.3 Interaction et Interface

Les Google Glass possèdent plusieurs moyens différents d'afficher l'interface graphique d'une application:

- Les live Cards : La Cards s'affiche sur la timeline dans les évènements passés. Elle est mise à jour périodiquement mais ne permet pas d'interactions poussées avec l'utilisateur.
- Static Cards (correspond plus au développement d'applications asynchrones, la réception de mails par exemple)
- L'immersion : Ce mode ressemble plus à la programmation classique sur Android. Une fois l'application lancée, les lunettes n'affiche plus que celle-ci. Les actions comme le glissement sur le touchpad par exemple entraînent une action dans l'application.
- Ongoing task: Un service en fond déclenchant l'affichage de vue et démarrant par événement (par exemple reconnaissance d'un réseau wifi pour la communication avec une base de données)

Pour notre programme nous avons choisi le mode immersion. En effet, le programme à but professionnel ne nécessite pas d'être quitté pour n'être regarder que ponctuellement.

#### 3.3.1 Reconnaissance vocale

La reconnaissance vocale de base sur Google Glass est fortement simplifiée avec le GDK. Nous avons assez facilement mis en oeuvre un menu vocal. Cependant beaucoup de mots sont difficiles à identifier, et pour les digits et les mots en une syllabe la reconnaissance est encore plus laborieuse.

Nous avons donc commencé à nous intéresser à un autre système de reconnaissance vocale pour palier à ces problèmes et pour permettre une reconnaissance vocale en français (offline).

La librairie PocketSphinx est l'une des plus utilisée en open-source, c'est donc celle que nous avons choisi d'implémenter. Pour supporter une langue, elle nécessite:

- Un modèle acoustique (ensemble d'informations pour le traitement sonore)

- Un dictionnaire associant les **mots** et leur *phonèmes* dont voici un extrait propre à notre application

```

i ii
j jj yy
j(2) jj ii
porte pp oo rr tt
porte(2) pp oo rr tt ee

```

- Une grammaire, c'est un fichier définissant comment seraient articulés les mots à reconnaître et limitant les possibilités de langage). Par exemple pour la reconnaissance vocale de la porte de départ, nous utilisons celle-ci.

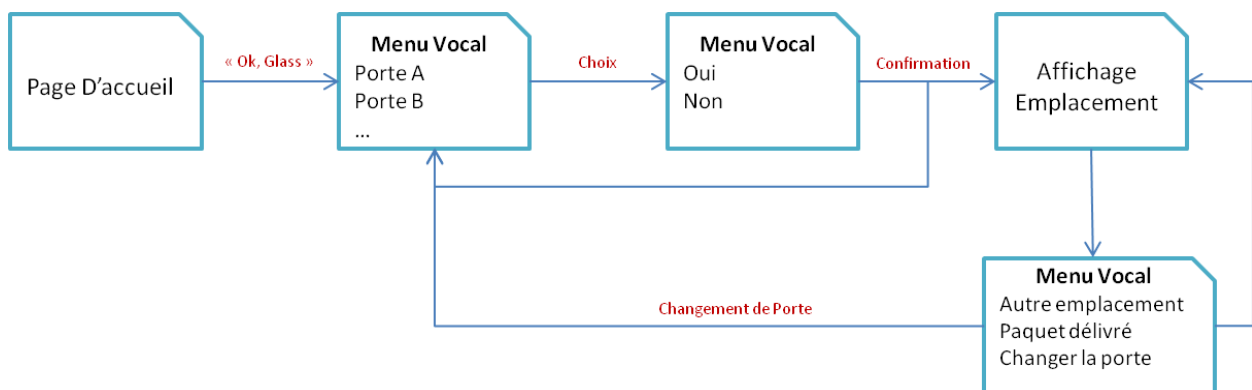
```

#JSGF V1.0;
grammar porte;
<porte> = porte a |
        porte b |
        porte c |
        porte d |
        porte e |
        porte f |
        porte g |
        porte h |
        porte i |
        porte j ;
public <porte> = <porte>+;

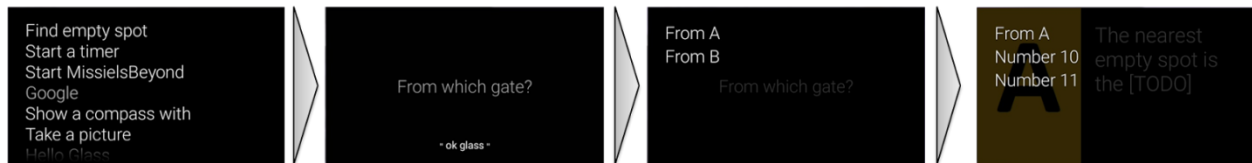
```

### 3.3.2 Scénario d'utilisation.

Nous avons donc programmé le logiciel prototype avec une reconnaissance vocale pour se déplacer dans les différents menus et afficher le résultat. Voici le logigramme de l'application:



Et voici les screenshots relatant d'une recherche d'emplacement:



=> Le plus dur dans cette partie aura été de comprendre le fonctionnement des lunettes. En effet, Google à crée un système de fonctionnement propre au Google Glass. Il est donc nécessaire de bien s'approprier ce fonctionnement avant de commencer à programmer.

## 4 Ce qui reste à faire

### 4.1 Développement des SmartGlass

Après avoir reçu la paire d'ORA de Optivent, nous devons réaliser le portage de l'application sur ces lunettes. Pour cela nous devons finir d'implémenter la reconnaissance vocale avec PocketSphinx sur les deux lunettes qui est déjà bien avancé et refaire l'interface qui est actuellement faite à l'aide du GDK. Cela permettrait aux utilisateurs de nous donner un comparatif entre les deux différents affichages de ces appareils.

La suite des évènements n'étant pas encore complètement définie à ce niveau nous allons voir avec le tuteur qu'elles seront les prochaines étapes de développement sur les lunettes. La suite pourrait s'orienter vers plusieurs possibilités.

- Retravailler l'application pour qu'elle corresponde mieux aux attentes des utilisateurs.
- Faire une communication avec une base de données fermée via Wifi.
  - Connaître le statut des emplacements.
  - Enregistrer les données relatives au magasinier

### 4.2 Projet d'affichage personnalisé TV

Notre tuteur nous a contactés pour nous proposer d'étudier un second projet en parallèle au projet initial. Celui consistant à gérer l'affichage d'information personnalisé d'un magasinier passant à proximité d'un écran de télévision. Nous avons effectué un Proof Of Concept (POC) et réfléchi à ce sujet. L'utilisation du BLE est donc de nouveau d'actualité, son utilité était discutable dans le projet initial. La question étant plus de savoir comment gérer le système du côté du téléviseur.

Le concept de base étant de :

- Fournir une beacon aux magasiniers de l'entrepôt
- Installer une télévision sur un centre de vie dans l'entrepôt avec un beacon
- Afficher du contenu personnalisé au magasinier qui passe devant la tv dans son centre de vie

Nous avons déjà fait le comparatif de plusieurs architectures et nous voyons avec notre tuteur laquelle nous allons utiliser pour mettre en place ce système.

## 5 Conclusion

### 5.1 Essai de l'application

Lors de notre 3<sup>ème</sup> rendez-vous nous avons pu faire tester notre première version de l'application. Les premiers retours venant de notre tuteur et de son supérieur furent positifs. Notre tuteur à ensuite récupéré les lunettes pour les faire essayer à d'autres personnes. Nous n'avons pas encore eu de retour précis mais l'interface semble plaire à la plupart des utilisateurs.

### 5.2 Bilan

Nous avons en deux mois réussi à prendre en main les SmartGlass et à rendre une première version de l'application. Nous trouvons que développer une interface réfléchiée en fonction de l'utilisateur est vraiment un point important et intéressant dans notre projet. Nous espérons pouvoir approfondir les fonctionnalités de ce programme. Ainsi que de faire un système complet et utilisable pour la télévision connectée. Nous attendons avec impatience la suite des évènements.